

APPLICATION
FOR
UNITED STATES LETTERS PATENT

INTERNATIONAL BUSINESS MACHINES CORPORATION

A METHOD OF LOGGING MESSAGE ACTIVITY

5 Field Of The Invention

10 The present invention relates, in general, to
messaging within a distributed data processing
environment and, in particular, to logging message
activity in such an environment.

15 Background to the Invention

20 Asynchronous transfer of messages between
application programs running on different data processing
systems within a network is well known in the art, and is
implemented by a number of commercially available
messaging systems. These systems include IBM
Corporation's MQSeries family of messaging products,
which use asynchronous messaging via queues. A sender
application program issues a PutMessage command to send
(put) a message to a target queue, and MQSeries queue
manager programs handle the complexities of transferring
the message under transactional control from the sender
25 to the target queue, which may be remotely located across
a heterogeneous computer network. The target queue is a
local input queue for another application program, which
retrieves (gets) the message from this input queue by
issuing a GetMessage command asynchronously from the send
30 operation. The receiver application program then performs

its processing on the message, and may generate further messages. MQSeries and IBM are trademarks of International Business Machines Corporation.

5 Transactional control of message transfer gives assured once and once-only message delivery of messages even in the event of system or communications failures. MQSeries products provide assured delivery by not finally deleting a message from storage on a sender system until
10 it is confirmed as safely stored by a receiver system, and by use of sophisticated recovery facilities. Prior to commitment of transfer of the message upon confirmation of successful storage, both the deletion of the message from storage at the sender system and insertion into
15 storage at the receiver system are kept 'in doubt' and can be backed out atomically in the event of a failure. This message transmission protocol and the associated transactional concepts and recovery facilities are
20 described in international patent application WO 95/10805 and US patent 5465328.

One key aspect of providing such transactional capabilities is the maintenance of a log in each system. The log, which may comprise one or more files, is used to
25 keep a track of completed message activity in the system. Each time a message is sent to a queue a record that the message was sent, including the message data, is written to the log, and each time a message is retrieved from a
30 queue a record that the message was retrieved is written to the log. Each of these writes to the log are forced to

disk (although some may be combined to a single force) because in the event of a failure the log is used to recover each queue to the state it was in at the point when the failure occurred. Such a failure could be, for example, due to a power loss causing immediate termination of the system. As a result, in order to provide such capabilities as once and once only delivery of messages, recovery cannot tolerate a log record being lost because, for example, it was buffered by the operating system at the point of failure.

Unfortunately however, forcing a log write to disk is a relatively slow operation and can have a significant impact on the performance of message delivery and retrieval. Further forcing a log write can be slower for larger writes and specifically when writing records relating to message sends which include the message data which is potentially large.

Summary Of the Invention

Accordingly, according to a first aspect the present invention provides a method for recording message activity in a log, the method comprising the steps of: receiving a request from an application to put a message, comprising message data, to a queue; and detecting whether there is a previous occurrence of the message data in the log, and if there is not a previous occurrence writing a log record including the message

data, but if there is a previous occurrence writing a log record including a reference for locating the previous occurrence of the message data in the log.

5 According to a second aspect the present invention provides a method for detecting the re-use of message data comprising the steps: receiving a request from an application to put a message, comprising message data, to a queue; and deducing, based on an indicator included with the request, that the message data was previously put to a message queue or got from a message queue by the application.

10 According to a third aspect the present invention provides a computer program comprising instructions which, when executed on a data processing host, causes said host to carry out a method of the first or the second aspect.

15 According to a fourth aspect the present invention provides a data processing apparatus comprising: a non-volatile memory storage device for storing log records thereon in a log comprising one or more log files; a volatile memory storage device; means for
20 receiving a request from an application to put a message, comprising message data, to a queue; means for detecting whether there is a previous occurrence of the message data in the log; means responsive to failing to detect a previous occurrence of the data in the log for writing a
25 log record including the message data; and means
30

responsive to detecting a previous occurrence of the data in the log for writing a log record including a reference for locating the previous occurrence of the message data in the log.

5

According to a fifth aspect the present invention provides a data processing apparatus comprising: means for receiving a request from an application to put a message, comprising message data, to a queue; and means for deducing, based on an indicator included with the request, that the message data was previously put to a message queue or got from a message queue by the application.

10

15

20

25

30

Thus the present invention reduces the size of selected records written to the log by a message processing system. When a message is put to a queue by an application a log record is written to the log which in the prior art includes the message data. However, according to the present invention, if the message data was included in a previous put and the message data from the previous put is available in the log, a reference to the previous occurrence of the message data in the log is included in the log record rather than the message data itself. As the message data is potentially large and the reference relatively small, less data is written to the log. Note that the previous put could be from the same application or a different application running on the same or a different data processing host.

Preferably the put request includes an indication that the message data was retrieved by the application in a previous request to get a message from the queue. This makes it easier to discover if the message is available in the log and this need only be done for message data that has previously been written to the log.

Preferably the put request includes an indication that the message data was included in a previous request, from the application, to put a message to a queue. This also makes it easier to discover if the message is available in the log.

Optionally the indication is a value which indicates that the message data was involved in the immediately preceding request from the application. For example it could indicate that the message data was also the message data included in an immediately preceding put request from the application. Further it could indicate that the message data was included in the message retrieved by the application in an immediately preceding get request. Note that the value could be a boolean value or, if it is required to know whether the immediately preceding request was a put or a get, a value (or values) comprising at least two bits.

Alternatively the indication is a token which uniquely identifies the message data within the scope of the application. This enables an application to identify message data that was involved in any preceding request.

For example, if the token is an integer, when the application first requests a message, with message data, to be put to a queue, the message data could be assigned the value 1. Next time the application wishes to put a message containing the same message data it can specify the value 1 to indicate that the message data was previously put. Similarly a token can be assigned by the messaging system on a get request.

Preferably in processing an application request to get a message from a queue the message processing system stores a reference, separate from the log and associated with the application, from which a previous occurrence of the message data can be found in the log. Preferably the reference is stored in volatile memory. This enables rapid access to the reference should the application subsequently request that a message, which includes the same message data, is put to a queue.

Preferably if, following a put request it is detected that there is no previous occurrence of message data in the log, in addition to writing a log record including the message data, a reference is stored, separate from the log and associated with the message, for subsequently locating the message data in the log. Preferably the reference is stored in volatile memory. This enables rapid access to the reference when a different application issues a get request to get the message.

Brief Description of the Drawings

The invention will now be described, by way of example only, with reference to a preferred embodiment thereof, as illustrated in the accompanying drawings, in which:

Figure 1 is a block diagram of data processing environment in which the preferred embodiment of the present invention is advantageously applied;

Figure 2 is a schematic diagram of typical log contents according to the prior art; and

Figure 3 is a schematic diagram of typical log contents according to the preferred embodiment of the present invention.

Figure 4 is a flow chart of the method for processing a putMessage request according to the preferred embodiment of the present invention.

Figure 5 is a flow chart of the method for processing a getMessage request according to the preferred embodiment of the present invention.

Note that in the figures, where a like part is included in more than one figure, where appropriate it is given the same reference number in each figure.

Detailed Description Of the Preferred Embodiment

10
15
20
25
30
In Fig. 1, a data processing host apparatus 10 is connected to other data processing host apparatuses 12 and 13 via a network 11, which could be, for example, the Internet. The hosts 10, 12 and 13, in the preferred embodiment, comprise messaging systems which cooperate to carry out the transfer of messages with guaranteed once and once only delivery. Although three hosts are shown, any number of similar hosts could be involved. Host 10 has a processor 101 for controlling the operation of the host 10, a RAM volatile memory element 102, a non-volatile memory element 103 on which a log of message activity is stored, and a network connector 104 for use in interfacing the host 10 with the network 11 to enable the hosts to communicate.

35
40
45
50
55
60
65
70
75
80
85
90
95
100
105
110
115
120
125
130
135
140
145
150
155
160
165
170
175
180
185
190
195
200
205
210
215
220
225
230
235
240
245
250
255
260
265
270
275
280
285
290
295
300
305
310
315
320
325
330
335
340
345
350
355
360
365
370
375
380
385
390
395
400
405
410
415
420
425
430
435
440
445
450
455
460
465
470
475
480
485
490
495
500
505
510
515
520
525
530
535
540
545
550
555
560
565
570
575
580
585
590
595
600
605
610
615
620
625
630
635
640
645
650
655
660
665
670
675
680
685
690
695
700
705
710
715
720
725
730
735
740
745
750
755
760
765
770
775
780
785
790
795
800
805
810
815
820
825
830
835
840
845
850
855
860
865
870
875
880
885
890
895
900
905
910
915
920
925
930
935
940
945
950
955
960
965
970
975
980
985
990
995
1000
1005
1010
1015
1020
1025
1030
1035
1040
1045
1050
1055
1060
1065
1070
1075
1080
1085
1090
1095
1100
1105
1110
1115
1120
1125
1130
1135
1140
1145
1150
1155
1160
1165
1170
1175
1180
1185
1190
1195
1200
1205
1210
1215
1220
1225
1230
1235
1240
1245
1250
1255
1260
1265
1270
1275
1280
1285
1290
1295
1300
1305
1310
1315
1320
1325
1330
1335
1340
1345
1350
1355
1360
1365
1370
1375
1380
1385
1390
1395
1400
1405
1410
1415
1420
1425
1430
1435
1440
1445
1450
1455
1460
1465
1470
1475
1480
1485
1490
1495
1500
1505
1510
1515
1520
1525
1530
1535
1540
1545
1550
1555
1560
1565
1570
1575
1580
1585
1590
1595
1600
1605
1610
1615
1620
1625
1630
1635
1640
1645
1650
1655
1660
1665
1670
1675
1680
1685
1690
1695
1700
1705
1710
1715
1720
1725
1730
1735
1740
1745
1750
1755
1760
1765
1770
1775
1780
1785
1790
1795
1800
1805
1810
1815
1820
1825
1830
1835
1840
1845
1850
1855
1860
1865
1870
1875
1880
1885
1890
1895
1900
1905
1910
1915
1920
1925
1930
1935
1940
1945
1950
1955
1960
1965
1970
1975
1980
1985
1990
1995
2000
2005
2010
2015
2020
2025
2030
2035
2040
2045
2050
2055
2060
2065
2070
2075
2080
2085
2090
2095
2100
2105
2110
2115
2120
2125
2130
2135
2140
2145
2150
2155
2160
2165
2170
2175
2180
2185
2190
2195
2200
2205
2210
2215
2220
2225
2230
2235
2240
2245
2250
2255
2260
2265
2270
2275
2280
2285
2290
2295
2300
2305
2310
2315
2320
2325
2330
2335
2340
2345
2350
2355
2360
2365
2370
2375
2380
2385
2390
2395
2400
2405
2410
2415
2420
2425
2430
2435
2440
2445
2450
2455
2460
2465
2470
2475
2480
2485
2490
2495
2500
2505
2510
2515
2520
2525
2530
2535
2540
2545
2550
2555
2560
2565
2570
2575
2580
2585
2590
2595
2600
2605
2610
2615
2620
2625
2630
2635
2640
2645
2650
2655
2660
2665
2670
2675
2680
2685
2690
2695
2700
2705
2710
2715
2720
2725
2730
2735
2740
2745
2750
2755
2760
2765
2770
2775
2780
2785
2790
2795
2800
2805
2810
2815
2820
2825
2830
2835
2840
2845
2850
2855
2860
2865
2870
2875
2880
2885
2890
2895
2900
2905
2910
2915
2920
2925
2930
2935
2940
2945
2950
2955
2960
2965
2970
2975
2980
2985
2990
2995
3000
3005
3010
3015
3020
3025
3030
3035
3040
3045
3050
3055
3060
3065
3070
3075
3080
3085
3090
3095
3100
3105
3110
3115
3120
3125
3130
3135
3140
3145
3150
3155
3160
3165
3170
3175
3180
3185
3190
3195
3200
3205
3210
3215
3220
3225
3230
3235
3240
3245
3250
3255
3260
3265
3270
3275
3280
3285
3290
3295
3300
3305
3310
3315
3320
3325
3330
3335
3340
3345
3350
3355
3360
3365
3370
3375
3380
3385
3390
3395
3400
3405
3410
3415
3420
3425
3430
3435
3440
3445
3450
3455
3460
3465
3470
3475
3480
3485
3490
3495
3500
3505
3510
3515
3520
3525
3530
3535
3540
3545
3550
3555
3560
3565
3570
3575
3580
3585
3590
3595
3600
3605
3610
3615
3620
3625
3630
3635
3640
3645
3650
3655
3660
3665
3670
3675
3680
3685
3690
3695
3700
3705
3710
3715
3720
3725
3730
3735
3740
3745
3750
3755
3760
3765
3770
3775
3780
3785
3790
3795
3800
3805
3810
3815
3820
3825
3830
3835
3840
3845
3850
3855
3860
3865
3870
3875
3880
3885
3890
3895
3900
3905
3910
3915
3920
3925
3930
3935
3940
3945
3950
3955
3960
3965
3970
3975
3980
3985
3990
3995
4000
4005
4010
4015
4020
4025
4030
4035
4040
4045
4050
4055
4060
4065
4070
4075
4080
4085
4090
4095
4100
4105
4110
4115
4120
4125
4130
4135
4140
4145
4150
4155
4160
4165
4170
4175
4180
4185
4190
4195
4200
4205
4210
4215
4220
4225
4230
4235
4240
4245
4250
4255
4260
4265
4270
4275
4280
4285
4290
4295
4300
4305
4310
4315
4320
4325
4330
4335
4340
4345
4350
4355
4360
4365
4370
4375
4380
4385
4390
4395
4400
4405
4410
4415
4420
4425
4430
4435
4440
4445
4450
4455
4460
4465
4470
4475
4480
4485
4490
4495
4500
4505
4510
4515
4520
4525
4530
4535
4540
4545
4550
4555
4560
4565
4570
4575
4580
4585
4590
4595
4600
4605
4610
4615
4620
4625
4630
4635
4640
4645
4650
4655
4660
4665
4670
4675
4680
4685
4690
4695
4700
4705
4710
4715
4720
4725
4730
4735
4740
4745
4750
4755
4760
4765
4770
4775
4780
4785
4790
4795
4800
4805
4810
4815
4820
4825
4830
4835
4840
4845
4850
4855
4860
4865
4870
4875
4880
4885
4890
4895
4900
4905
4910
4915
4920
4925
4930
4935
4940
4945
4950
4955
4960
4965
4970
4975
4980
4985
4990
4995
5000
5005
5010
5015
5020
5025
5030
5035
5040
5045
5050
5055
5060
5065
5070
5075
5080
5085
5090
5095
5100
5105
5110
5115
5120
5125
5130
5135
5140
5145
5150
5155
5160
5165
5170
5175
5180
5185
5190
5195
5200
5205
5210
5215
5220
5225
5230
5235
5240
5245
5250
5255
5260
5265
5270
5275
5280
5285
5290
5295
5300
5305
5310
5315
5320
5325
5330
5335
5340
5345
5350
5355
5360
5365
5370
5375
5380
5385
5390
5395
5400
5405
5410
5415
5420
5425
5430
5435
5440
5445
5450
5455
5460
5465
5470
5475
5480
5485
5490
5495
5500
5505
5510
5515
5520
5525
5530
5535
5540
5545
5550
5555
5560
5565
5570
5575
5580
5585
5590
5595
5600
5605
5610
5615
5620
5625
5630
5635
5640
5645
5650
5655
5660
5665
5670
5675
5680
5685
5690
5695
5700
5705
5710
5715
5720
5725
5730
5735
5740
5745
5750
5755
5760
5765
5770
5775
5780
5785
5790
5795
5800
5805
5810
5815
5820
5825
5830
5835
5840
5845
5850
5855
5860
5865
5870
5875
5880
5885
5890
5895
5900
5905
5910
5915
5920
5925
5930
5935
5940
5945
5950
5955
5960
5965
5970
5975
5980
5985
5990
5995
6000
6005
6010
6015
6020
6025
6030
6035
6040
6045
6050
6055
6060
6065
6070
6075
6080
6085
6090
6095
6100
6105
6110
6115
6120
6125
6130
6135
6140
6145
6150
6155
6160
6165
6170
6175
6180
6185
6190
6195
6200
6205
6210
6215
6220
6225
6230
6235
6240
6245
6250
6255
6260
6265
6270
6275
6280
6285
6290
6295
6300
6305
6310
6315
6320
6325
6330
6335
6340
6345
6350
6355
6360
6365
6370
6375
6380
6385
6390
6395
6400
6405
6410
6415
6420
6425
6430
6435
6440
6445
6450
6455
6460
6465
6470
6475
6480
6485
6490
6495
6500
6505
6510
6515
6520
6525
6530
6535
6540
6545
6550
6555
6560
6565
6570
6575
6580
6585
6590
6595
6600
6605
6610
6615
6620
6625
6630
6635
6640
6645
6650
6655
6660
6665
6670
6675
6680
6685
6690
6695
6700
6705
6710
6715
6720
6725
6730
6735
6740
6745
6750
6755
6760
6765
6770
6775
6780
6785
6790
6795
6800
6805
6810
6815
6820
6825
6830
6835
6840
6845
6850
6855
6860
6865
6870
6875
6880
6885
6890
6895
6900
6905
6910
6915
6920
6925
6930
6935
6940
6945
6950
6955
6960
6965
6970
6975
6980
6985
6990
6995
7000
7005
7010
7015
7020
7025
7030
7035
7040
7045
7050
7055
7060
7065
7070
7075
7080
7085
7090
7095
7100
7105
7110
7115
7120
7125
7130
7135
7140
7145
7150
7155
7160
7165
7170
7175
7180
7185
7190
7195
7200
7205
7210
7215
7220
7225
7230
7235
7240
7245
7250
7255
7260
7265
7270
7275
7280
7285
7290
7295
7300
7305
7310
7315
7320
7325
7330
7335
7340
7345
7350
7355
7360
7365
7370
7375
7380
7385
7390
7395
7400
7405
7410
7415
7420
7425
7430
7435
7440
7445
7450
7455
7460
7465
7470
7475
7480
7485
7490
7495
7500
7505
7510
7515
7520
7525
7530
7535
7540
7545
7550
7555
7560
7565
7570
7575
7580
7585
7590
7595
7600
7605
7610
7615
7620
7625
7630
7635
7640
7645
7650
7655
7660
7665
7670
7675
7680
7685
7690
7695
7700
7705
7710
7715
7720
7725
7730
7735
7740
7745
7750
7755
7760
7765
7770
7775
7780
7785
7790
7795
7800
7805
7810
7815
7820
7825
7830
7835
7840
7845
7850
7855
7860
7865
7870
7875
7880
7885
7890
7895
7900
7905
7910
7915
7920
7925
7930
7935
7940
7945
7950
7955
7960
7965
7970
7975
7980
7985
7990
7995
8000
8005
8010
8015
8020
8025
8030
8035
8040
8045
8050
8055
8060
8065
8070
8075
8080
8085
8090
8095
8100
8105
8110
8115
8120
8125
8130
8135
8140
8145
8150
8155
8160
8165
8170
8175
8180
8185
8190
8195
8200
8205
8210
8215
8220
8225
8230
8235
8240
8245
8250
8255
8260
8265
8270
8275
8280
8285
8290
8295
8300
8305
8310
8315
8320
8325
8330
8335
8340
8345
8350
8355
8360
8365
8370
8375
8380
8385
8390
8395
8400
8405
8410
8415
8420
8425
8430
8435
8440
8445
8450
8455
8460
8465
8470
8475
8480
8485
8490
8495
8500
8505
8510
8515
8520
8525
8530
8535
8540
8545
8550
8555
8560
8565
8570
8575
8580
8585
8590
8595
8600
8605
8610
8615
8620
8625
8630
8635
8640
8645
8650
8655
8660
8665
8670
8675
8680
8685
8690
8695
8700
8705
8710
8715
8720
8725
8730
8735
8740
8745
8750
8755
8760
8765
8770
8775
8780
8785
8790
8795
8800
8805
8810
8815
8820
8825
8830
8835
8840
8845
8850
8855
8860
8865
8870
8875
8880
8885
8890
8895
8900
8905
8910
8915
8920
8925
8930
8935
8940
8945
8950
8955
8960
8965
8970
8975
8980
8985
8990
8995
9000
9005
9010
9015
9020
9025
9030
9035
9040
9045
9050
9055
9060
9065
9070
9075
9080
9085
9090
9095
9100
9105
9110
9115
9120
9125
9130
9135
9140
9145
9150
9155
9160
9165
9170
9175
9180
9185
9190
9195
9200
9205
9210
9215
9220
9225
9230
9235
9240
9245
9250
9255
9260
9265
9270
9275
9280
9285
9290
9295
9300
9305
9310
9315
9320
9325
9330
9335
9340
9345
9350
9355
9360
9365
9370
9375
9380
9385
9390
9395
9400
9405
9410
9415
9420
9425
9430
9435
9440
9445
9450
9455
9460
9465
9470
9475
9480
9485
9490
9495
9500
9505
9510
9515
9520
9525
9530
9535
9540
9545
9550
9555
9560
9565
9570
9575
9580
9585
9590
9595
9600
9605
9610
9615
9620
9625
9630
9635
9640
9645
9650
9655
9660
9665
9670
9675
9680
9685
9690
9695
9700
9705
9710
9715
9720
9725
9730
9735
9740
9745
9750
9755
9760
9765
9770
9775
9780
9785
9790
9795
9800
9805
9810
9815
9820
9825
9830
9835
9840
9845
9850
9855
9860
9865
9870
9875
9880
9885
9890
9895
9900
9905
9910
9915
9920
9925
9930
9935
9940
9945
9950
9955
9960
9965
9970
9975
9980
9985
9990
9995
10000
10005
10010
10015
10020
10025
10030
10035
10040
10045
10050
10055
10060
10065
10070
10075
10080
10085
10090
10095
10100
10105
10110
10115
10120
10125
10130
10135
10140
10145
10150
10155
10160
10165
10170
10175
10

extent, is of a fixed size and is used to store log records, in chronological order, relating to message activity for one or more queues. When an extent file becomes full a new extent is opened, each extent being numbered sequentially. Periodically a maintenance operation, known as checkpointing, is performed in which extents that no longer contain information required for recovery are made redundant and can be deleted. Any log record written to a log file that has not been marked redundant is available to be read and the position of each log record is known by the extent number of the extent in which it is contained and an offset within that extent. Note that in other embodiments, for example, the log could comprise a database or one or more circular files.

Fig 2, shows schematically an example of the typical contents of a log, in a prior art messaging system, for a specified sequence of requests. Note that the requests in the sequence may be issued by one or more applications and represent only one of the possible sequences of requests received by a messaging system. Note also that the format of the requests shown is merely illustrative. The first request puts a message, comprising control information "a" and data "A", to a message queue. Details of this message are then written to the log in a log record (201, 202) which comprises two elements, the control information (201) and the data (202) as specified by the application. Message control information is relatively small and can include such information as a

message id and chaining information which will be processed by the receiving application. Message data is the content of the message to be sent and is potentially very large. Note that in the figs. 2 and 3 the relative sizes of the log records and record elements are not to scale. Further log records may contain one or more other elements, for example to allow navigation of the log, which are not shown.

The second request puts a message, comprising control information "b" and data "B", to a message queue. This may or may not be the same message queue as the first message. The log record (203, 204) for the put of this message comprises similar information as the log record written for the put of the first message, and in fact similar information as the log record written for the put of any message, namely the message control information and data. The third request retrieves the previously put message with data "A", and as a result a record (205) is written to the log to record this fact. Note that this type of record is relatively small as it contains no application data. The next two requests are a put and get of a message comprising control information "c" and data "C", which result in a large log record (206, 207) associated with the put and a small log record (208) associated with the get. The next request is a put of a message comprising control information "d" and data "A". This request may be from the same application that previously put the message with data "A" or the application that previously got the message with data

"A". Either way a log record (209,210) is written to the log similar to the previous records associated with the put of a message. The final two requests get messages with data "B" and "A" respectively resulting in two small log records (211, 212) to note the fact.

Fig 3, shows schematically an example of the typical contents of a log file, in the preferred embodiment of the present invention, for the sequence of requests shown for Figure 2. For the first 5 requests the log contents are the same as Fig. 2. However, when the message comprising control information "d" and data "A" is put to a queue the contents of the log diverge from those of Fig. 2. The put of this message results in a log record (209, 301) being written to the log which contains the control information "d" specified with the put request and a reference, depicted by arrow 302, to the previous log record (201, 202) which contains the message data "A". Note that the reference comprises the extent number and offset. As a result log record (209, 301) does not include the message data "A" where equivalent log record (209, 210) in Fig. 2 does, and therefore potentially much less information, depending on the size of the message data, is written to the log. The remainder of the contents of the log are the same as for Fig. 2.

Note that the reference (301) recorded in the log may, for example, refer to the start of the log record (201,202) containing the data. Alternatively it could refer to a position, such as the position of the data

(202) within the log record (201,202). Further, if the same message data is included in a series of messages such that it is put and got more than once, the reference (301) may refer to a record within a chain of one or more records that ends with the record containing the message data.

In the example shown in Fig. 3 the amount of log space saved, and therefore the performance improvement gained, may not appear to be very significant. However it is relatively common for an application to get a message from an input queue and put a message containing the same data to one or more output queues. For example Publish/Subscribe is commonly used in this way. As a result the improved performance and saved log space can be significant to such applications.

In order to use the method of writing information to the log file as described for Fig. 3, the messaging system must be able to recognize that a message being put contains data that has previously been put and therefore has already been written to the log. There are many ways of doing this. The method employed in the preferred embodiment requires a flag which is added to the putMessage request by an application and indicates whether the message being put contains the same message data as the previous message got or put by the application. This method is illustrated in figures 4 and 5. In other embodiments the messaging system could, for example, scan the log for the data or save in storage an

abbreviated form of message data, such as a hash value, with which the message data specified on a put request can be compared.

5 Figure 4 shows the processing of a put request according to the preferred embodiment of the present invention. The processing of a message containing data that has not previously been put in a message will now be described with reference to Fig. 4. At step 401 a
10 putMessage request is received from an application. At step 402 a check is made for an indication on the putMessage request that the message data is the same as that of the previous put or get by the application. In
15 this scenario this indication is not set and processing continues to step 404 where a record is written to the log containing details of the message, including the message data. At step 405 a reference, comprising a
20 position in the log (extent number and offset), to the log record just written is saved in volatile storage and associated with the message and the application. This enables the position of the log record containing the data to be obtained, without accessing the log, both
25 during processing of a get request for the message even if the request is received from a different application and during a second put request by the same application. Finally at step 407 the message is added to the queue specified in the request.

30 Figure 5 shows the processing of a get request according to the preferred embodiment of the present

invention. At step 501 a getMessage request is received from an application. At step 502 a check is made to see if the position in the log of the log record containing the message data is known. This will have been stored and associated with the message at step 404 of Fig. 4.

However, as some messages may remain in a queue for a long period, the position of the log record previously stored may have been removed from volatile storage based on a maintenance algorithm which is not part of the present invention. If the position of the log record is known it is, at step 503, associated with the application, which may require its duplication in volatile storage. Whether or not the position of the log record was known processing of the getMessage request completes at step 504 where a record is written to the log to indicate that the message has been retrieved and step 505 where the message is returned to the requester (i.e.: the application that issued the getMessage request).

The processing of a putMessage request containing message data that has previously been put in a message will now be described with reference to Fig. 4. At step 401 a putMessage request is received from an application. At step 402 a check is made for an indication on the putMessage request that the message data is the same as that of the previous put or get by the application. In this scenario this indication is set and processing continues to step 403 where a check is made to see if the position of the log record containing the message data is

known and is available. It will be known if its position in the log was previously stored in volatile storage, and associated with the application, at either step 405 in Fig. 4 or step 503 of Fig. 5, and this has not been subsequently been removed from volatile storage by a maintenance operation. It will be available if the log record is still available in the log. The log record may not be available, for example, if a message data is re-used a long time after it was originally logged and the extent file in which it was written has been made redundant as part of a completed checkpoint operation, or is scheduled to be made redundant once an in-progress checkpoint operation has completed. If the position of the log record is known and is still available in the log, a log record is written to the log which comprises the message control information and a reference, comprising a position in the log (extent number and offset), to the log record that contains the message data. However if the position of the log record is not known or the log record is not available processing continues with steps 404 and 405. At step 404 a record comprising the message control information and data is written to the log. At step 405 the position, in the log, of the log record just written is saved in volatile storage and associated with the message and the application. Note that step 405 is not executed after step 406 so that if message data is included in a series of messages such that, for example, if it is put and got more than once, the message data does not have to be accessed through a chain of log records. The method

completes, following steps 405 and 406, by adding the message to the queue specified in the request at step 407.

5 Note that the method of Figure 4 may be carried out in more than one host in the case where a putMessage request is received on a given data processing host to place a message on a queue in a remote data processing host. As a result for a single message the steps of
10 Figure 4 may be carried out in two hosts where some steps are performed on both hosts and other steps are performed on just one of the hosts. For example step 401 is likely to be only carried out on the host on which the request is received and step 407 only on the host on
15 which the queue exists whereas all other steps are likely to be carried out on both hosts although this will be implementation dependent.

20 Thus the preferred embodiment of the invention has been described whereby a log record may be written to the log, as part of the processing a put request, that does not include the message data but a reference to a previous occurrence of the message data in the log. Although the preferred embodiment carries this out for
25 messages that contain the same data when they are either put in consecutive requests or put immediately after get, in other embodiments only one of these options may be implemented.

Further, in another embodiment, message data could be associated with a reference unique to the data and within the requesting application. This would be assigned by the application on a put request and by the messaging system on a get request. This would allow a subsequent put request to specify the reference in order to indicate that the data had previously been put or got by the application and therefore written to the log. This would remove the restriction in the preferred embodiment that, for example, a get request must be immediately followed by a put request with the same data in order to take advantage of the invention.

Further, it is possible that message data is duplicated in storage other than a log used to track message activity. As a result the methods disclosed in the present invention for detecting re-use of message data for the purpose of reducing the amount of data written to the log could be used in isolation for reducing duplication of the message data in other areas of storage.